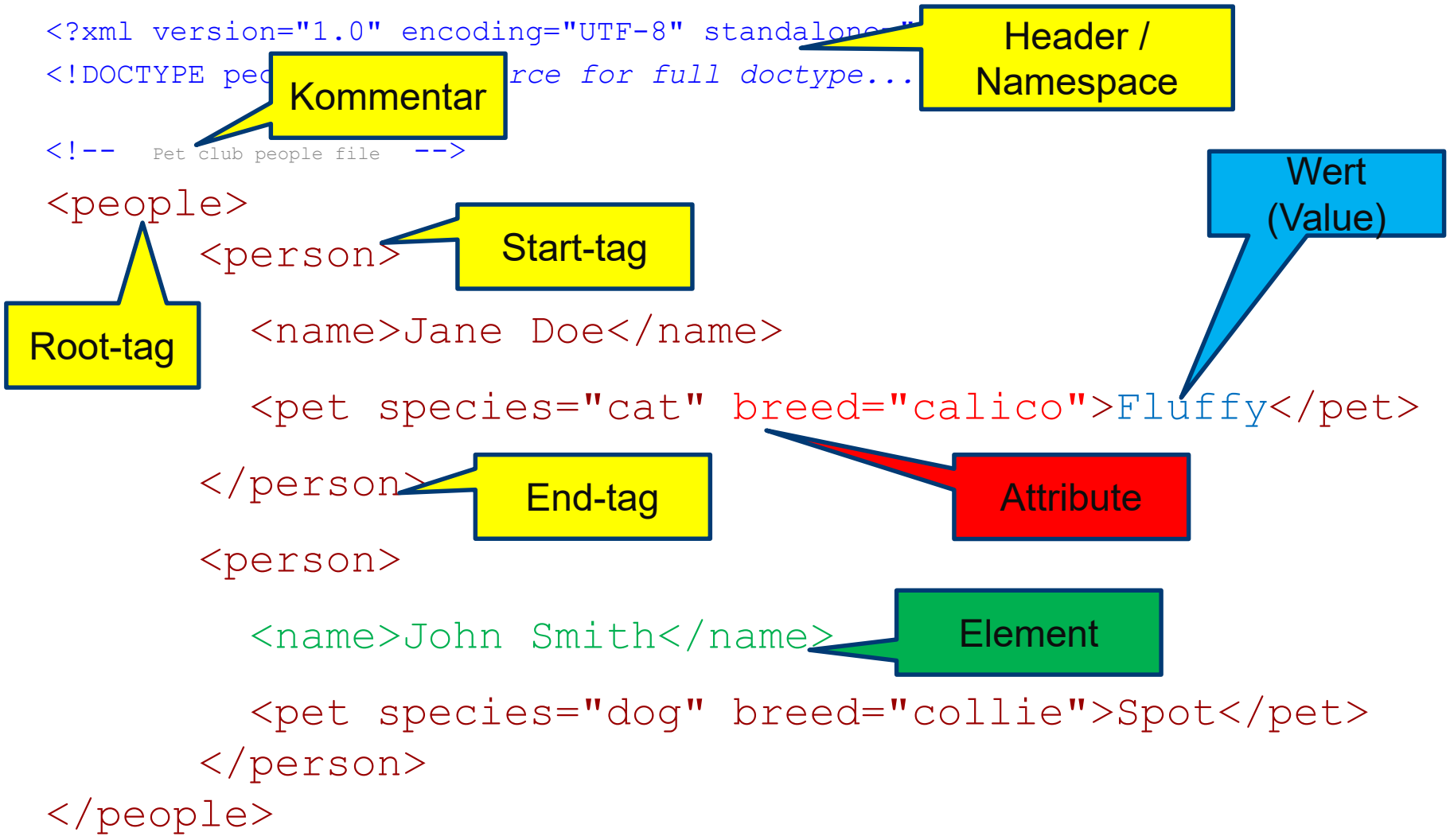
XML (Extended Markup Language)

Version: 6.11.20

XML-Basics (Extensible Markup Language)



XML: Correct

Syntax Regeln (Wohlgeformtheit, Correct)

- Textstring
- Nur ein root tag
- Tags korrekt geschlossen und nicht überlappend
- Attribut-Werte in Anführungszeichen
- Gross- / Kleinschreibung wird unterschieden
- UNICODE

Struktur definieren (Gültigkeit, Valid)

DTD (Document Type Definition)

```
<!ELEMENT betreff (#PCDATA)>
<!ELEMENT body (#PCDATA | betreff | datum)*>
<!ELEMENT datum (#PCDATA)>
```

XML Schema (XSD) → Selber XML Document

```
<xs:element name="Destination">
 <xs:complexType>
   <xs:simpleContent>
     <xs:extension base="xs:string">
           <xs:attribute name="node" type="xs:string" use="required"/>
           <xs:attribute name="user" type="xs:string" use="required"/>
```

XML: Navigieren (Daten lesen) Parsen

Zwei Standard-Parser Typen sind vorhanden

DOM → Document Object Model

Document wird als ganzes geparsed und in eine Daten-Struktur gelegt

SAX → Simple API for XML

Event gesteuert

XML: Navigieren (Daten lesen) Parsen

➔ XPath ist eine Query Language für XML

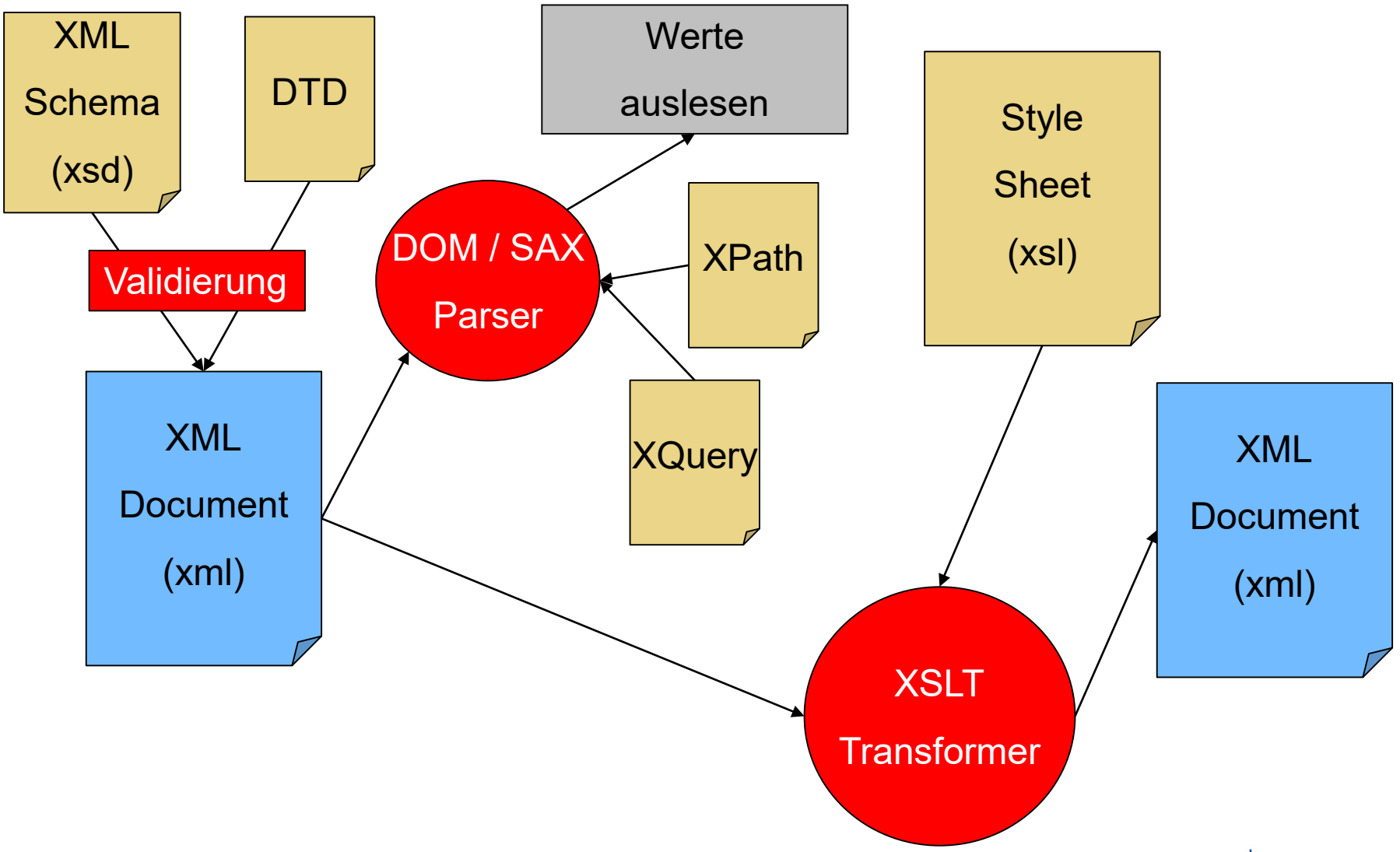
Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. **Note:** IE5 and later has implemented that [0] should be the first node, but according to the W3C standard it should have been [1]!!
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='eng']	Selects all the title elements that have an attribute named lang with a value of 'eng'
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

XML Transformation (XSLT ist Teil von XSL)

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="employee">
        <p>
        <xsl:value-of select="firstName[@home='Peterliwiese']"/>
        </p>
    </xsl:template>
</xsl:stylesheet>
```

XHTML: HTML, dass auch ein XML Dokument ist ➔ XHTML
Strenge Regeln (z.B. statt
 ➔
)

XML Zusammenfassung



Cascaded Style Sheet (CSS)

Cascaded Style Sheet (CSS)

HTML ist eine Markup-Language.

➔ Definiert nur Struktur des Dokuments (nicht Layout)

➔ Mit CSS können Elemente genau positioniert oder formatiert werden

Einbinden von CSS

```
<head>
 <title>Das style-Attribut</title>
 <link rel="stylesheet"                          href="formate.css">
 <link rel="stylesheet" media="screen"           href="website.css">
 <link rel="stylesheet" media="print, embossed" href="druck.css">
 <link rel="stylesheet" media="aural"            href="speaker.css">
 <style type="text/css">
 /* ... hier sind dateispezifische Formate erlaubt ... */
 </style>
 </head>
```

http://de.wikipedia.org/wiki/Cascading_Style_Sheets

http://de.selfhtml.org/css/index.htm

Cascaded Style Sheet (CSS)

Beispiel eines CSS für HTML-Elemente

```
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin black; }
```

Gilt fuer diese Elemente. Unterelemente

```
  * { color:blue; }
```

Universalselektor; gilt fuer alle Elemente

```
  p,li { font-size:110%; line-height:140%; font-family:Helvetica,Arial,sans-serif; word-spacing:0.3em; }
```

Eigenschaften gelten für <p> **und**

```
  h1 i { color:blue; font-style:normal; }
```

Eigenschaften gelten nur für <i> **innerhalb von** <h1>

```
  div * b { color:violet; }
```

Eigenschaften gelten nur für <b> **innerhalb (beliebig tief) von**

```
</style>
```

Cascaded Style Sheet (CSS)

Beispiel eines CSS mit attributbedingten Formaten

```
<style type="text/css">
    p[align] { color:red; }
```

Gilt für <p> bei denen das Attribute **align** gesetzt ist

```
    p[align=center] { color:blue; text-align:left; }
```

Gilt für <p> bei denen das Attribute **align** auf **center** gesetzt ist

```
    td[abbr~=Berlin] { background-color:#FFFF00 }
```

Gilt für <td> bei denen das Attribute **abbr** den Text **Berlin** enthaelt

```
   *[lang|=en] { background-color:#FF0000; color:#FFFFFF; }
```

Gilt für alle Elemente bei denen das Attribute **lang** mit dem Text **en** beginnen

```
</style>
```

Cascaded Style Sheet (CSS)

Beispiel eines CSS und deren Anwendung (class)

```
<style type="text/css">
 .beitrag { border:1px outset gray; margin:.5em; padding:.5em; background-color:#efd;}
 .uebersetzung { border-bottom:1px dotted #900; }
 .autor { font-style:italic; }
 .datum { font-size:80%; color:#444; }
 h1.hinterlegt { background-color:#FFFF00 }
 </style>
```

Unterschied und :

- Beide tags relativ eigenschaftslos
- Neue Zeile
- kein neuer Absatz (im Textfluss)

```
<div class="beitrag">
 <p>Beispielsweise in der
   <a href="http://www.w3.org/TR/REC-html40/">HTML 4.01-
   <span lang="en" title="Empfehlung" class="uebersetzung">Recommendation</span></a>.
 </p>
 <p class="autor">Max Hilfreich, <span class="datum">22.05.2006</span> </p>
 <h1 class="hinterlegt">H1 knallgelb hinterlegt</h1>
</div>
```

Cascaded Style Sheet (CSS)

Beispiel eines CSS und deren Anwendung (id)

```
<style type="text/css">
 #roterBereich {
    position:absolute; top:130px; left:30px; width:320px;
    padding:10px; margin:0px; border:4px solid #EE0000;
 }
 </style>
```

```
<div id="roterBereich"><h1>Der rote Bereich</h1></div>
```

Layout / CSS: http://www.dynamicdrive.com/style/